Package ‘latticeDensity’

April 18, 2021

Type Package

Title Density Estimation and Nonparametric Regression on Irregular Regions

Version 1.2.6

Date 2021-04-14

Author Ronald Barry <`rpbarry@alaska.edu`>

Maintainer Ronald Barry <`rpbarry@alaska.edu`>

Depends R (>= 3.5.0), spatstat (>= 2.0-0), spatstat.geom

Imports splancs, spdep, spam, stats, utils, graphics, grDevices, sp, spatialreg, sf

LazyData true

Suggests knitr, devtools, roxygen2, testthat, rmarkdown, rgdal

Encoding UTF-8

Description Functions that compute the lattice-based density estimator of Barry and McIntyre, which accounts for point processes in two-dimensional regions with irregular boundaries and holes. The package also implements two-dimensional non-parametric regression for similar regions.

License GPL-2

URL www.r-project.org

Repository CRAN

Date/Publication 2021-04-18 16:20:02 UTC

RoxygenNote 7.1.1

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

`addObservations` *Input observations for use in the lattice-based density estimator*

Description

This function takes a `formLatticeOutput` object, which encodes a region possibly with and irregular boundary and holes, and adds point process observations. The observations should be in the form of a matrix or data frame. `addObservations` is used when the aim is to produce a density map from a point process. If, instead, you wish to produce a nonparametric regression surface given responses and their locations, you should use `addQuantVar` instead.

Usage

```
addObservations(formLatticeOutput, observations)
```

Arguments

`formLatticeOutput`	An object returned by formLattice or editLattice.
`observations`	A matrix or data frame with two columns.

Details

Every node in the `formLatticOutput` object is assigned an initial density that is equal to the fraction of all observations that are nearest to that node. Note that this means observations can be outside the boundary of the region of interest - they will just be associated with the nearest node inside the region. The function returns a vector equal in length to the number of nodes that has the initial density for each node. This vector corresponds to p_0, the inital probability vector as in Barry and McIntyre (2011).

Value

a list with two elements.

- init_probNumerical vector with the initial probability distribution
- which_nodesvector of nodes to which observations were assigned

Author(s)

Ronald P. Barry

References

Ronald P. Barry, Julie McIntyre. Estimating animal densities and home range in regions with irregular boundaries and holes: A lattice-based alternative to the kernel density estimator. Ecological Modelling 222 (2011) 1666-1672. <doi:10.1016/j.ecolmodel.2011.02.016>

Examples

```
plot.new()
data(polygon1)
#
nodeFillingOutput <- nodeFilling(poly=polygon1, node_spacing=0.01)
plot(nodeFillingOutput)
formLatticeOutput <- formLattice(nodeFillingOutput)
plot(formLatticeOutput)
#
Pointdata <- splancs::csr(polygon1,30)
Pointdata <- Pointdata[Pointdata[,1]<0.5,]
colnames(Pointdata) <- c("x","y")
plot(polygon1,type="n")
polygon(polygon1)
points(Pointdata,pch=19)
#
densityOut <- createDensity(formLatticeOutput,PointPattern=Pointdata,
                            k=40,intensity=FALSE, sparse = TRUE)
plot(densityOut)
```

`addQuantVar` *Input data for Nonparametric Regression smoothing.*

Description

This function takes a `formLatticeOutput` object, which encodes a region possibly with and irregular boundary and holes. This and a matrix of locations where a response variable has been measured, and a vector of the responses, is used to create an initial distribution for use in the non-parametric regression function `createNparReg`. If, instead, you have a point process and wish to produce a density estimate, you should use the function `addObservations`.

Usage

```
addQuantVar(formLatticeOutput, Z, locations)
```

Arguments

`formLatticeOutput`	An object from the functions formLattice or editLattice.
`Z`	A vector of response variable values.
`locations`	A two-column matrix or data frame of data locations. • init_quantvar Vector of initial quantitative variables • init_prob Vector of initial probability density • which_nodes What nodes are closest to each data location #' @references Ronald P. Barry, Julie McIntyre. Estimating animal densities and home range in regions with irregular boundaries and holes: A lattice-based alternative to the kernel density estimator. Ecological Modelling 222 (2011) 1666-1672. <doi:10.1080/10618600.2017.1375935>

References

Julie McIntyre, Ronald P. Barry (2018) A Lattice-Based Smoother for Regions with Irregular Boundaries and Holes. Journal of Computational and Graphical Statistics. In Press.

`areaRegion` *Computes area of a region*

Description

This function computes the area of a region by first finding the area of the bounding polygon, then subtracting the area of each hole.

Usage

```
areaRegion(formLatticeOutput)
```

Arguments

`formLatticeOutput`	An object returned by formLattice or editLattice.

Value

Numeric The area of the bounded region.

Warning

Note that this program does not check to see if the holes are non-intersecting or if the holes intersect the polygon.

Examples

```
data(areaRegionExample)
attach(areaRegionExample)
hole_list <- list(hole1,hole2)
nodeFillingOutput <- nodeFilling(poly=boundary, node_spacing=0.03,
                              hole_list = hole_list)
formLatticeOutput <- formLattice(nodeFillingOutput)
plot(formLatticeOutput)
areaRegion(formLatticeOutput)
```

`areaRegionExample` *A region with two holes*

Description

Three 2-column matrices. The first is a set of vertices of a boundary polygon encompassing a region. The other two, hole1 and hole2, are holes in the region. This example dataset is used to illustrate the situation where there are holes in the region where a density or nonparametric regression is to be applied. It is used in the example of the function `areaRegion`.

Usage

```
areaRegionExample
```

Format

Three 2-column numerical matrices

`createDensity` *Generates a density using random walks on a lattice.*

Description

Given a lattice and a point pattern of observations, createDensity starts random walks at each observation. k steps are taken and the output is a densityOut object, which can be used to plot a density estimate. If you wish to perform non-parametric regression, you should use the functions addQuantVar and createNparReg instead.

Usage

```
createDensity(
  formLatticeOutput,
  PointPattern = NULL,
  M = 0.5,
  k,
  intensity = FALSE,
  sparse = TRUE
)
```

Arguments

`formLatticeOutput`	An object from formLattice or editLattice.
`PointPattern`	A 2-column matrix or data frame of locations.
`M`	Maximum probability of random walk moving.
`k`	The smoothing parameter (number of steps).
`intensity`	Plot an intensity vs a density.
`sparse`	If TRUE, matrix computations are sparse.

Details

We start with an initial probability density p0 where the ith entry in p0 is the fraction of the point pattern that is nearest to the ith node. This is the empirical density function with no smoothing. If T is the transition matrix, and given a number of steps in the diffusion, T k p0 is the probability density of the diffusion after k steps. This is the major output of this function, along with information needed to produce a plot, including the polygons for the boundary and holes, and a vector of NS coordinates and EW coordinates used by the contour function. All of the necessary information for plotting is bundled in the object of class densityOutLBDE. Details of this process can be found in Barry and McIntyre (2011).

Value

An object of type densityOut

- EW_locs A vector of EW coordinates of nodes.
- NS_locs A vector of NS coordinates of nodes.
- boundaryPoly The boundary of the region (two-columns).
- hole_list A list of polygonal holes in the region.
- PointPattern A 2-column matrix of observations.
- probs The probability distribution over the nodes.
- densityLBDE Density in a form for making a contour map.
- area The area of the region, with holes removed.

Author(s)

Ronald P. Barry

References

Ronald P. Barry, Julie McIntyre. Estimating animal densities and home range in regions with irregular boundaries and holes: A lattice-based alternative to the kernel density estimator. Ecological Modelling 222 (2011) 1666-1672. <doi:10.1080/10618600.2017.1375935>

Examples

```
plot.new()
data(polygon1)
nodeFillingOutput <- nodeFilling(poly=polygon1, node_spacing=0.02)
plot(nodeFillingOutput)
formLatticeOutput <- formLattice(nodeFillingOutput)
plot(formLatticeOutput)
Pointdata <- splancs::csr(polygon1,75)
Pointdata <- Pointdata[Pointdata[,1]<0.5,]
plot(polygon1,type="n")
polygon(polygon1)
points(Pointdata,pch=19)
out <- crossvalDensity(formLatticeOutput,PointPattern=Pointdata,
                       M=0.5,max_steps = 35)
densityOut <- createDensity(formLatticeOutput,
                            PointPattern=Pointdata,
                            k=out$k,intensity=FALSE, sparse = TRUE)
plot(densityOut)
homerange(densityOut, percent = 0.95)
```

`createNparReg` *Performs nonparametric regression on irregular regions.*

Description

This function takes the lattice from formLattice (which fills the region of interest) along with the list of responses and their locations, and creates a prediction surface. The approach is kernel nonparametric regression with the kernels created by a k-step diffusion on the lattice about each location where a response was collected.

Usage

```
createNparReg(formLatticeOutput, Z, PointPattern = NULL, M = 0.5, k)
```

Arguments

`formLatticeOutput`	An object returned by formLattice or editLattice.
`Z`	Vector of responses to be smoothed.
`PointPattern`	A 2 column matrix or data frame of locations.
`M`	The maximum probability that the random walk will move.
`k`	Number of steps.

Details

We denote by $K_{ik}(s)$ the kernel obtained by assigning the node nearest to the ith response and then running a k-step diffusion on the lattice and evaluating the resulting density at location s. Then the estimator $\hat{f}(s) = (\sum_i K_{ik}(s) * Z_i)/\sum_i K_{ik}(s)$ which is the traditional kernal regression estimator with diffusion kernels. This approach leads to a non-parametric regression that respects the boundaries of the polygonal region. The construction of the kernels is detailed in Barry and McIntyre (2011). Using kernels to perform nonparametric regression is described in many publications, including Wasserman (2006).

Value

A list of class NparRegOut with elements:

- EW_locs Vector of EW locations.
- NS_locs Vector of NS locations.
- nodes Matrix of node locations in lattice.
- boundaryPoly Matrix showing bounding polynomial.
- hole_list List of polygons, holes in region.
- PointPattern Matrix of the locations of the data.
- which_nodes Matrix locations of nodes closest to data.
- NparRegNum Vector of numerators of the regression estimates
- NparRegDenom Vector of denominators of the regression estimates
- sigma2 Numeric, estimate of the noise variance.

Variance Estimation

We use the variance estimator $\sum e_{i,-i}^2/n$, where $e_{i,-i}$ is the ith deleted residual.

References

Larry Wasserman. All of Nonparametric Statistics. Springer Science + Business Media, Inc. N.Y. 2006.

#' @references Julie McIntyre, Ronald P. Barry (2018) A Lattice-Based Smoother for Regions with Irregular Boundaries and Holes. Journal of Computational and Graphical Statistics. <doi:10.1080/10618600.2017.1375935>

Examples

```
data(nparExample)
attach(nparExample)
plot.new()
#  Simulate a response variable
index1 = (grid2[,2]<0.8)|(grid2[,1]>0.6)
Z = rep(NA,length(grid2[,1]))
n1 = sum(index1)
n2 = sum(!index1)
Z[index1] <- 3*grid2[index1,1] + 4 + rnorm(n1,0,sd=0.4)
Z[!index1] <- -2*grid2[!index1,1] + 4 + rnorm(n2,0,sd=0.4)
#
coords=rbind(polygon2,polygon2[1,])
plot(coords,type="l")
points(grid2,pch=19,cex=0.5,xlim=c(-0.1,1))
text(grid2,labels=round(Z,1),pos=4,cex=0.5)
#
nodeFillingOutput <- nodeFilling(poly=polygon2, node_spacing=0.025)
plot(nodeFillingOutput)
formLatticeOutput <- formLattice(nodeFillingOutput)
NparRegOut <- createNparReg(formLatticeOutput,Z,PointPattern=grid2,k=2)
plot(NparRegOut)
```

`crossvalDensity` *UBC crossvalidation for the lattice-based density estimator.*

Description

A function to perform crossvalidation to determine the smoothing parameter for the lattice-based density estimator. It minimizes the UCV criterion.

Usage

```
crossvalDensity(
  formLatticeOutput,
  PointPattern,
```

```
  M = 0.5,
  max_steps = 200,
  sparse = TRUE
)
```

Arguments

`formLatticeOutput`	An object from formLattice or editLattice.
`PointPattern`	A matrix or data frame of locations.
`M`	The maximum probability that the random walk will move.
`max_steps`	The maximum number of steps attempted.
`sparse`	Whether spare matrix computations used.

Details

The function computes the k-step diffusion $p_k = T^k p_0$, then computes the Unbiased CrossValidation (UCV) criterion of Sain, Baggerly and Scott (1994). This function can compute the UCV using either full matrix methods or sparse (default) matrix methods. The latter are almost always much faster, though it is possible that if the number of points in the point pattern is large compared to the number of nodes (an unlikely circumstance) that the full matrix method would be quicker. The sparse matrix approach typically uses less memory. The paper by Barry and McIntyre (2010) shows the approximation to the UCV used in this approach.

Value

- ucv The value of the goodness-of-fit statistic.
- k The number of steps.

References

Crossvalidation of Multivariate Densities. Stephan R. Sain, Keith A. Baggerly, David W. Scott; Journal of the American Statistical Association, Vol. 89 (1994) 807-817

Julie McIntyre, Ronald P. Barry (2018) A Lattice-Based Smoother for Regions with Irregular Boundaries and Holes. Journal of Computational and Graphical Statistics. <doi:10.1080/10618600.2017.1375935>

Examples

```
plot.new()
data(polygon1)
#
nodeFillingOutput <- nodeFilling(poly=polygon1, node_spacing=0.02)
plot(nodeFillingOutput)
formLatticeOutput <- formLattice(nodeFillingOutput)
plot(formLatticeOutput)
#
Pointdata <- splancs::csr(polygon1,75)
Pointdata <- Pointdata[Pointdata[,1]<0.5,]
plot(polygon1,type="n")
```

```
polygon(polygon1)
points(Pointdata,pch=19)
#
out <- crossvalDensity(formLatticeOutput,PointPattern=Pointdata,
       M=0.5,max_steps = 70)
#
densityOut <- createDensity(formLatticeOutput,PointPattern=Pointdata,
                          k=out$k,intensity=FALSE, sparse = TRUE)
plot(densityOut)
#
homerange(densityOut, percent = 0.95)
```

`crossvalNparReg` *Crossvalidation for non-parametric regression.*

Description

Performs least-squares crossvalidation for the lattice-based non-parametric regression estimator.

Usage

```
crossvalNparReg(formLatticeOutput, Z, PointPattern, M = 0.5, max_steps = 200)
```

Arguments

`formLatticeOutput`	An object from formLattice or editLattice.
`Z`	Vector of response values to be smoothed.
`PointPattern`	A 2-column matrix or data frame of locations.
`M`	Maximum probability that the random walk moves.
`max_steps`	Maximum number of steps attempted.

Details

For a given k, deleted residuals are computed for each of the observations. The crossvalidation is based on minimization of the squares of the deleted residuals.

Value

A list consisting of

- SumSq Vector of crossvalidated sums of squares
- Number of steps that minimizes the crossvalidated SS.

Author(s)

Ronald P. Barry

References

Ronald P. Barry, Julie McIntyre. Estimating animal densities and home range in regions with irregular boundaries and holes: A lattice-based alternative to the kernel density estimator. Ecological Modelling 222 (2011) 1666-1672.

Julie McIntyre, Ronald P. Barry (2018) A Lattice-Based Smoother for Regions with Irregular Boundaries and Holes. Journal of Computational and Graphical Statistics. <doi:10.1080/10618600.2017.1375935>

Examples

```
data(nparExample)
attach(nparExample)
plot.new()
#  Simulate a response variable
index1 <- (grid2[,2]<0.8)|(grid2[,1]>0.6)
Z <- rep(NA,length(grid2[,1]))
n1 <- sum(index1)
n2 <- sum(!index1)
Z[index1] <- 3*grid2[index1,1] + 4 + rnorm(n1,0,sd=0.4)
Z[!index1] <- -2*grid2[!index1,1] + 4 + rnorm(n2,0,sd=0.4)
#
plot(polygon2,type="n")
polygon(polygon2)
points(grid2,pch=19,cex=0.5,xlim=c(-0.1,1))
text(grid2,labels=round(Z,1),pos=4,cex=0.5)
#
nodeFillingOutput <- nodeFilling(poly=polygon2, node_spacing=0.025)
plot(nodeFillingOutput)
formLatticeOutput <- formLattice(nodeFillingOutput)
plot(formLatticeOutput)
hold <- crossvalNparReg(formLatticeOutput,Z,
                       PointPattern=grid2,M=0.5,max_steps = 40)
NparRegOut <- createNparReg(formLatticeOutput,Z,PointPattern=grid2,k=hold$k)
plot(NparRegOut)
```

`deletedResid` *Deleted residuls for non-parametric regression.*

Description

Computes deleted residuals for the lattice-based non-parametric regression estimator.

Usage

```
deletedResid(formLatticeOutput, Z, PointPattern, M = 0.5, k)
```

Arguments

`formLatticeOutput`	An object from formLattice or editLattice.
`Z`	Vector of response values.
`PointPattern`	2-column matrix or data frame of locations.
`M`	Maximum probability that the random walk moves.
`k`	Number of steps in random walk.

Value

A vector of deleted residuals.

Author(s)

Ronald P. Barry

References

Ronald P. Barry, Julie McIntyre. Estimating animal densities and home range in regions with irregular boundaries and holes: A lattice-based alternative to the kernel density estimator. Ecological Modelling 222 (2011) 1666-1672.

Julie McIntyre, Ronald P. Barry (2018) A Lattice-Based Smoother for Regions with Irregular Boundaries and Holes. Journal of Computational and Graphical Statistics. <doi:10.1080/10618600.2017.1375935>

Examples

```
data(nparExample)
attach(nparExample)
plot.new()
#  Simulate a response variable
index1 <- (grid2[,2]<0.8)|(grid2[,1]>0.6)
Z <- rep(NA,length(grid2[,1]))
n1 <- sum(index1)
n2 <- sum(!index1)
Z[index1] <- 3*grid2[index1,1] + 4 + rnorm(n1,0,sd=0.4)
Z[!index1] <- -2*grid2[!index1,1] + 4 + rnorm(n2,0,sd=0.4)
#
plot(polygon2,type="n")
polygon(polygon2)
points(grid2,pch=19,cex=0.5,xlim=c(-0.1,1))
text(grid2,labels=round(Z,1),pos=4,cex=0.5)
#
nodeFillingOutput <- nodeFilling(poly=polygon2, node_spacing=0.025)
plot(nodeFillingOutput)
formLatticeOutput <- formLattice(nodeFillingOutput)
plot(formLatticeOutput)
hold <- crossvalNparReg(formLatticeOutput,Z,
                        PointPattern=grid2,M=0.5,max_steps = 40)
deletedResid(formLatticeOutput,Z,
```

```
            PointPattern=grid2,M=0.5,k=hold$k)
```

`editLattice` *Add or remove links in the lattice*

Description

editLattice is an interactive editor based on the function edit.nb from the package spdep. A formLatticeOutput object includes an automatically generated neighborhood structure. Occasionally this will either leave two nodes disconnected that should be connected or vice versa. editLattice allows the user to directly edit the plot of the lattice using mouseclicks to add or remove neighbor links between nodes.

Usage

```
editLattice(formLatticeOutput)
```

Arguments

`formLatticeOutput`	An object from formLattice or editLattice.

Value

a formLatticeOutput object, which contains

- EWlocs EW coordinates for use in contour.
- NSlocs NS coordinates for use in contour.
- nodes Matrix of node locations.
- poly Matrix of vertices of the boundary polygon.
- latt Lattice object as generated by dnearneigh of package spdep.

Author(s)

Ronald P. Barry

See Also

formLattice

Examples

```
## Not run:
plot.new()
data(polygon1)
nodeFillingOutput = nodeFilling(poly=polygon1, node_spacing=0.03)
plot(nodeFillingOutput)
formLatticeOutput = formLattice(nodeFillingOutput)
plot(formLatticeOutput)
formLatticeOutput = editLattice(formLatticeOutput)
#
#  Paste the code above into R, then do the editing before
#  pasting the code below into R.
#
Pointdata = splancs::csr(polygon1,20)
densityOut = createDensity(formLatticeOutput,PointPattern=Pointdata,
                    k=150,intensity=FALSE, sparse = TRUE)
plot(densityOut)

## End(Not run)
```

`formLattice` *Builds a neighbor structure on the nodes.*

Description

formLattice connects all nodes into a neighbor lattice by linking any two nodes that are within 1.5*node_spacing. Typically this will result in links in the E, W, N, S, NE, NW, SE, SW directions. The lattice object is created by the function dnearneigh from spdep.

Usage

```
formLattice(nodeFillingOutput)
```

Arguments

`nodeFillingOutput`	An object, as produced by the function nodeFilling.

Details

When forming the lattice, the function does not check to see if any node is completely isolated from the rest of the nodes, nor does it check to see that paths exist between all pairs of nodes. Thus the lattice might be disconnected. You can still determine a nonparametric density in this case, but you need to think about whether it makes sense to allow disconnected sublattices. If you wish to connect isolated nodes to the lattice, use the editing function editLattice.

Value

formLatticeOutput object

- EW_locs EW coordinates for use by contour
- NS_locs NS coordinates for use by contour
- nodes Matrix of node locations.
- poly Outer boundary.
- latt Neighbor lattice.
- hole.poly List of hole polygons.

Author(s)

Ronald P. Barry

References

Ronald P. Barry, Julie McIntyre. Estimating animal densities and home range in regions with irregular boundaries and holes: A lattice-based alternative to the kernel density estimator. Ecological Modelling 222 (2011) 1666-1672. <doi:10.1016/j.ecolmodel.2011.02.016>

Examples

```
plot.new()
data(polygon1)
nodeFillingOutput <- nodeFilling(poly=polygon1, node_spacing=0.02)
plot(nodeFillingOutput)
formLatticeOutput <- formLattice(nodeFillingOutput)
plot(formLatticeOutput)
Pointdata <- splancs::csr(polygon1,80)
Pointdata <- Pointdata[Pointdata[,1]<0.5,]
plot(polygon1,type="n")
polygon(polygon1)
points(Pointdata,pch=19)
densityOut <- createDensity(formLatticeOutput,PointPattern=Pointdata,
                            k=20,intensity=FALSE, sparse = TRUE)
plot(densityOut)
homerange(densityOut, percent = 0.95)
```

`homerange` *Produces a homerange map.*

Description

homerange produces a map of the homerange, for any given percentage. The homerange contains the smallest number of nodes with total density greater than the percent. This function is illustrated in Barry and McIntyre (2011).

Usage

```
homerange(densityOut, percent = 0.95, output = FALSE)
```

Arguments

`densityOut`	A densityOut object, produced by createDensity.
`percent`	the sum of the probabilities of all nodes in the homerange exceeds this value.
`output`	if TRUE, the function returns a matrix containing, for each node, a location (first two columns) and whether the node is in the homerange.

Value

A list of two vectors used for mapping:

- nodes The coordinates of all nodes in the model
- ind Indicator functions, is the location in the homerange?

Author(s)

Ronald P. Barry

References

Ronald P. Barry, Julie McIntyre. Estimating animal densities and home range in regions with irregular boundaries and holes: A lattice-based alternative to the kernel density estimator. Ecological Modelling 222 (2011) 1666-1672. <doi:10.1016/j.ecolmodel.2011.02.016>

Examples

```
    plot.new()
data(polygon1)
nodeFillingOutput <- nodeFilling(poly=polygon1, node_spacing=0.015)
plot(nodeFillingOutput)
formLatticeOutput <- formLattice(nodeFillingOutput)
plot(formLatticeOutput)
Pointdata <- splancs::csr(polygon1,75)
Pointdata <- Pointdata[Pointdata[,1]<0.5,]
plot(polygon1,type="n")
polygon(polygon1)
points(Pointdata,pch=19)
out <- crossvalDensity(formLatticeOutput,PointPattern=Pointdata,
                     M=0.5,max_steps = 40)
densityOut <- createDensity(formLatticeOutput,PointPattern=Pointdata,
k=out$k,intensity=FALSE, sparse = TRUE)
plot(densityOut)
homerange(densityOut, percent = 0.95)
```

`makeTmatrix` *Create the transition matrix for the diffusion.*

Description

This function generates a transition matrix for the diffusion process on the lattice.

Usage

```
makeTmatrix(formLatticeOutput, M = 0.5, sparse = TRUE)
```

Arguments

`formLatticeOutput`	A formLatticeOutput object, returned by the functions formLattice or by the function editLattice.
`M`	A smoothing parameter. It is the maximum probability that the random walk moves from the node in a single step. It is a maximum probability in the sense that this is the movement probability for nodes not near a boundary. Of course, near a boundary movement will be constrained proportional to how many neighbors the node has. Thus if interior nodes have eight neighbors, a node with only four neighbors will move half as often. Since the number of steps k also determines smoothing, M is usually left at 0.5. Note that values of M=1 or M=0 can lead to pathological results. The paper of Barry and McIntyre (2011) shows the exact construction of the transition matrix.
`sparse`	logical. If TRUE, then uses sparse matrix computations from packages spdep and spam. If FALSE, uses full matrix computations. The use of sparse matrices is almost always more efficient.

Value

An NxN transition matrix, where N is the number of nodes.

Author(s)

Ronald P. Barry

References

Ronald P. Barry, Julie McIntyre. Estimating animal densities and home range in regions with irregular boundaries and holes: A lattice-based alternative to the kernel density estimator. Ecological Modelling 222 (2011) 1666-1672. <doi:10.1016/j.ecolmodel.2011.02.016>

Examples

```
plot.new()
data(polygon1)
nodeFillingOutput <- nodeFilling(poly=polygon1, node_spacing=0.02)
formLatticeOutput <- formLattice(nodeFillingOutput)
Pointdata <- splancs::csr(polygon1,75)
Pointdata <- Pointdata[Pointdata[,1]<0.5,]
poly.area <- splancs::areapl(polygon1)
init_prob <- addObservations(formLatticeOutput, Pointdata)
T = makeTmatrix(formLatticeOutput, M = 0.5, sparse=TRUE)
p10 <- Tkp(T, 10, p=init_prob$init_prob)
head(cbind(init_prob$init_prob, p10))
```

`nodeFilling` *Produces a grid of locations inside the region boundary.*

Description

nodeFilling produces a grid of locations that are the nodes in the diffusion process.

Usage

```
nodeFilling(poly, node_spacing, hole_list = NULL)
```

Arguments

`poly`	A matrix that contains the vertices of the bounding polygon.
`node_spacing`	The distance between grid locations.
`hole_list`	Optional list of holes to be removed from the region

Details

nodeFilling superimposes a square grid of points over the region, with spacing given by the parameter node_spacing. The points contained in the region are retained. The output, a nodeFillingOutput object, contains the boundaries of the region (and holes), the set of nodes, and EW and NS coordinates necessary for creating a contour plot.

Value

An object of type nodeFillingOutput is produced.

- EW_locs EW coordinates for the contour plot.
- NS_locs NS coordinates for the contour plot.
- nodes Matrix of node locations.
- poly Matrix of vertices of boundary polygon.
- node_spacing Vertical and horizontal node spacing.
- hole_list List of polygons representing holes in region.

Author(s)

Ronald P. Barry

References

Ronald P. Barry, Julie McIntyre. Estimating animal densities and home range in regions with irregular boundaries and holes: A lattice-based alternative to the kernel density estimator. Ecological Modelling 222 (2011) 1666-1672. <doi:10.1016/j.ecolmodel.2011.02.016>

Examples

```
plot.new()
data(polygon1)
nodeFillingOutput <- nodeFilling(poly=polygon1,node_spacing=0.02)
plot(nodeFillingOutput)
formLatticeOutput <- formLattice(nodeFillingOutput)
plot(formLatticeOutput)
```

`nparExample` *Example boundary with a grid of locations*

Description

The first item, `polygon2`, is 35x2 matrix describing the boundary of a region. The second, `grid2`, is a set of 59 locations for simulated values of a response variable. The third item, `Z`, is a vector of responses. This dataset was created to test and illustrate the non-parametric lattice based regression estimator. See the example for function `createNparReg`.

Usage

```
nparExample
```

Format

Two matrices and a vector. One matrix is 35x2, the other is 59x2.

References

Julie McIntyre, Ronald P. Barry (2018) A Lattice-Based Smoother for Regions with Irregular Boundaries and Holes. Journal of Computational and Graphical Statistics. In Press.

`plot.densityOut` *Plot the density map*

Description

Plots the boundary, all holes and the locations of all nodes along with the density contour map.

Usage

```
## S3 method for class 'densityOut'
plot(x, ...)
```

Arguments

`x`	An object of type densityOut returned by createDensity.
`...`	Graphical parameters for the function contour.default.

Author(s)

Ronald P. Barry

References

Ronald P. Barry, Julie McIntyre. Estimating animal densities and home range in regions with irregular boundaries and holes: A lattice-based alternative to the kernel density estimator. Ecological Modelling 222 (2011) 1666-1672

Examples

```
plot.new()
data(polygon1)
#
nodeFillingOutput <- nodeFilling(poly=polygon1, node_spacing=0.025)
plot(nodeFillingOutput)
formLatticeOutput <- formLattice(nodeFillingOutput)
plot(formLatticeOutput)
#
Pointdata <- splancs::csr(polygon1,75)
Pointdata <- Pointdata[Pointdata[,1]<0.5,]
plot(polygon1,type="n")
polygon(polygon1)
points(Pointdata,pch=19)
#
densityOut <- createDensity(formLatticeOutput,PointPattern=Pointdata,
                            k=55,intensity=FALSE, sparse = TRUE)
plot(densityOut)
#
homerange(densityOut, percent = 0.95)
```

`plot.formLatticeOutput`

Plot the lattice.

Description

This function plots the boundary, holes, nodes and neighbor lattice for the lattice based density or regression estimators. The plot can be examined to determine whether the lattice of connected nodes fills the region. If some nodes are connected when they should not be, or are disconnected when they should be connected, use editLattice to add or remove neighbor links.

Usage

```
## S3 method for class 'formLatticeOutput'
plot(x, ...)
```

Arguments

`x`	An object of type formLatticeOutput returned by either formLattice or editLattice.
`...`	Other arguments to be passed to functions plot, points, lines.

Author(s)

Ronald P. Barry

Examples

```
plot.new()
data(polygon1)
nodeFillingOutput <- nodeFilling(poly=polygon1, node_spacing=0.015)
plot(nodeFillingOutput)
formLatticeOutput <- formLattice(nodeFillingOutput)
plot(formLatticeOutput)
```

`plot.nodeFillingOutput`

Plot a nodeFillingOutput object.

Description

Plots the boundary, all holes and the locations of all nodes. Should be used to decide if the nodes fill the region and are spaced closely enough to give good resolution in the plots. The only reason not to make the nodes too closely spaced is when the computing time or memory becomes too great.

Usage

```
## S3 method for class 'nodeFillingOutput'
plot(x, ...)
```

Arguments

`x`	An object of type nodeFillingOutput returned by either nodeFilling or removeHole.
`...`	Other arguments to be passed to functions plot, points, lines.

Author(s)

Ronald P. Barry

References

Ronald P. Barry, Julie McIntyre. Estimating animal densities and home range in regions with irregular boundaries and holes: A lattice-based alternative to the kernel density estimator. Ecological Modelling 222 (2011) 1666-1672.

Examples

```
plot.new()
data(polygon1)
nodeFillingOutput <- nodeFilling(poly=polygon1, node_spacing=0.01)
plot(nodeFillingOutput)
```

`plot.NparRegOut` *Plot the non-parametric regression surface.*

Description

Takes as input a NparRegOut object from the function createNparReg. This plotting function makes a countour plot of the non-parametric regression prediction surface.

Usage

```
## S3 method for class 'NparRegOut'
plot(x, ...)
```

Arguments

`x`	An object of type NparRegOut returned by createNparReg.
`...`	Other arguments to be passed to functions plot, points, lines.

Author(s)

Ronald P. Barry

References

Julie McIntyre, Ronald P. Barry (2018) A Lattice-Based Smoother for Regions with Irregular Boundaries and Holes. Journal of Computational and Graphical Statistics. In Press.

Examples

```
data(nparExample)
attach(nparExample)
plot.new()
#  Simulate a response variable
index1 <- (grid2[,2]<0.8)|(grid2[,1]>0.6)
Z <- rep(NA,length(grid2[,1]))
n1 <- sum(index1)
n2 <- sum(!index1)
Z[index1] <- 3*grid2[index1,1] + 4 + rnorm(n1,0,sd=0.4)
Z[!index1] <- -2*grid2[!index1,1] + 4 + rnorm(n2,0,sd=0.4)
#
plot(polygon2,type="n")
polygon(polygon2)
points(grid2,pch=19,cex=0.5,xlim=c(-0.1,1))
text(grid2,labels=round(Z,1),pos=4,cex=0.5)
#  Following is the generation of the nonparametric
#  regression prediction surface.
nodeFillingOutput <- nodeFilling(poly=polygon2, node_spacing=0.025)
plot(nodeFillingOutput)
formLatticeOutput <- formLattice(nodeFillingOutput)
plot(formLatticeOutput)
NparRegOut <- createNparReg(formLatticeOutput,Z,PointPattern=grid2,k=2)
plot(NparRegOut)
```

`plot.varianceMapOut` *Plot the standard error map.*

Description

Takes as input a varianceMapOut object from the function varianceMap. This plotting function makes a countour plot of the non-parametric regression standard error surface.

Usage

```
## S3 method for class 'varianceMapOut'
plot(x, ...)
```

Arguments

`x`	An object of type varianceMapOut returned by varianceMap.
`...`	Other arguments to be passed to functions plot, points, lines.

Author(s)

Ronald P. Barry

References

Julie McIntyre, Ronald P. Barry (2018) A Lattice-Based Smoother for Regions with Irregular Boundaries and Holes. Journal of Computational and Graphical Statistics. In Press.

Examples

```
data(nparExample)
attach(nparExample)
plot.new()
#  Simulate a response variable
index1 <- (grid2[,2]<0.8)|(grid2[,1]>0.6)
Z <- rep(NA,length(grid2[,1]))
n1 <- sum(index1)
n2 <- sum(!index1)
Z[index1] <- 3*grid2[index1,1] + 4 + rnorm(n1,0,sd=0.4)
Z[!index1] <- -2*grid2[!index1,1] + 4 + rnorm(n2,0,sd=0.4)
#
plot(polygon2,type="n")

polygon(polygon2)
points(grid2,pch=19,cex=0.5,xlim=c(-0.1,1))
text(grid2,labels=round(Z,1),pos=4,cex=0.5)
#
nodeFillingOutput <- nodeFilling(poly=polygon2, node_spacing=0.025)
plot(nodeFillingOutput)
formLatticeOutput <- formLattice(nodeFillingOutput)
plot(formLatticeOutput)
hold <- crossvalNparReg(formLatticeOutput,Z,
                       PointPattern=grid2,M=0.5,max_steps = 25)
var_map <- varianceMap(formLatticeOutput,Z,
             PointPattern=grid2,M=0.5,k=hold$k)
plot(var_map)
```

---

polygon1 *Example boundary with causeway*

Description

A 2x19 matrix of vertices for the boundary of a region representing a lake almost divided in half by a causeway. This was used in a simulation in the paper of Barry and McIntyre (2011).

Usage

```
polygon1
```

Format

A 2x19 numerical matrix

Source

Ronald P. Barry, Julie McIntyre. Estimating animal densities and home range in regions with irregular boundaries and holes: A lattice-based alternative to the kernel density estimator. Ecological Modelling 222 (2011) 1666-1672.

`predict.NparRegOut` *Predictions at data locations from lattice-based non-parametric regression.*

Description

Takes as input a NparRegOut object from the function createNparReg. A vector of predicted values is produced corresponding to each location in the data.

Usage

```
## S3 method for class 'NparRegOut'
predict(object, new_pred = NULL, ...)
```

Arguments

`object`	An object of type NparRegOut returned by createNparReg.
`new_pred`	if new_pred is left out, predictions are made at the locations of the point pattern. Otherwise, new_pred is a 2-column matrix of locations where you wish to obtain predictions.
`...`	Aditionally arguments affecting the predictions, of which there are none at this time.

Details

If new_pred is not used as an arguments, this function returns a vector of predictions at each node closest to an observations of the original point process. If you wish to make predictions at arbitrary locations, let new_pred be a 2-column matrix of those locations. Note that all predictions are actually at the nearest node to the desired locations. NOTE: Like all functions in this package, new locations are relocated to the nearest node in the region, even if they are outside the boundary. Thus you should ensure that your locations of interest are inside the boundary and that the density of nodes is high enough that the nearest node is close enough to the location you queried.

Value

Vector of predicted values.

Author(s)

Ronald P. Barry

References

Julie McIntyre, Ronald P. Barry (2018) A Lattice-Based Smoother for Regions with Irregular Boundaries and Holes. Journal of Computational and Graphical Statistics. <doi:10.1080/10618600.2017.1375935>

Examples

```
data(nparExample)
attach(nparExample)
plot.new()
#  Simulate a response variable
index1 <- (grid2[,2]<0.8)|(grid2[,1]>0.6)
Z <- rep(NA,length(grid2[,1]))
n1 <- sum(index1)
n2 <- sum(!index1)
Z[index1] <- 3*grid2[index1,1] + 4 + rnorm(n1,0,sd=0.4)
Z[!index1] <- -2*grid2[!index1,1] + 4 + rnorm(n2,0,sd=0.4)
#
plot(polygon2,type="n")
polygon(polygon2)
points(grid2,pch=19,cex=0.5,xlim=c(-0.1,1))
text(grid2,labels=round(Z,1),pos=4,cex=0.5)
#  Following is the generation of the nonparametric
#  regression prediction surface.
nodeFillingOutput <- nodeFilling(poly=polygon2, node_spacing=0.025)
plot(nodeFillingOutput)
formLatticeOutput <- formLattice(nodeFillingOutput)
plot(formLatticeOutput)
NparRegOut <- createNparReg(formLatticeOutput,Z,PointPattern=grid2,k=2)
plot(NparRegOut)
predict(NparRegOut)
```

`removeHole` *Removes holes from the region prior to density estimation.*

Description

If a hole in a region is specified as a polygon, the function removeHole removes all nodes in the nodeFillingOutput that are contained in the hole. This function is called by nodeFilling, so it is generally not needed by users.

Usage

```
removeHole(hole_poly, nodeFillingOutput)
```

Arguments

`hole_poly`	A numerical matrix of vertices of the hole polygon.
`nodeFillingOutput`	An object of type nodeFillingOutput, returned by nodeFilling or removeHole.

Value

An object of type nodeFillingOutput, with values:

- EW_locs EW coordinates for the contour plot.
- NS_locs NS coordinates for the contour plot.
- nodes Matrix of node locations.
- poly Matrix of vertices of boundary polygon.
- node_spacing Vertical and horizontal node spacing.
- hole_list List of polygons representing holes in region.

Author(s)

Ronald P. Barry

`Tkp` *Compute the vector T^k*p*

Description

TranMat is the transition matrix of the random walk on the lattice. By multiplying by the probability density p at time t, you get the probability density at time t+1. Thus, to get the probability density after k steps, pk, compute pk = Tkp1. This application of finite Markov processes is described in Barry and McIntyre (2011).

Usage

```
Tkp(TranMat, k, p)
```

Arguments

`TranMat`	Transition matrix returned by makeTmatrix.
`k`	The number of steps in the diffusion.
`p`	A numerical vector of length equal to the number of nodes, of initial probabilities.

Value

A vector of probabilities.

Author(s)

Ronald P. Barry

References

Ronald P. Barry, Julie McIntyre. Estimating animal densities and home range in regions with irregular boundaries and holes: A lattice-based alternative to the kernel density estimator. Ecological Modelling 222 (2011) 1666-1672.

Examples

```
plot.new()
data(polygon1)
nodeFillingOutput <- nodeFilling(poly=polygon1, node_spacing=0.015)
formLatticeOutput <- formLattice(nodeFillingOutput)
Pointdata <- splancs::csr(polygon1,75)
Pointdata <- Pointdata[Pointdata[,1]<0.5, ]
init_prob <- addObservations(formLatticeOutput, Pointdata)
TranMat <- makeTmatrix(formLatticeOutput, M = 0.5, sparse=TRUE)
p10 <- Tkp(TranMat, k=10, p=init_prob$init_prob)
head(cbind(init_prob$init_prob, p10))
```

`varianceMap` *Spatial variance for the regression smoother.*

Description

Computes the variance at each location for the non-parametric regression estimator.

Usage

```
varianceMap(formLatticeOutput, Z, PointPattern, M = 0.5, k)
```

Arguments

`formLatticeOutput`	An object from formLattice or editLattice.
`Z`	Vector of response values.
`PointPattern`	2-column matrix or data frame of locations.
`M`	Maximum probability that the random walk moves.
`k`	Number of steps in random walk.

Details

`varianceMap` computes an estimated variance at each node in the lattice, output in a form for mapping with contour. The approach is the Nadaraya-Watson kernel variance estimator: $s^2 \sum K^2(si, s0)/(\sum K(si, s0))^2$. It's important to note that this should not be overused as a prediction error, as kernel estimators are not unbiased.

Value

VarianceMapOut object

- EW_locs EW coordinates for use by contour
- NS_locs NS coordinates for use by contour
- boundaryPoly vertices of the boundary
- hole_list list of polygonal hole boundaries, if any.
- SE_map_grid estimated standard error at each location

Author(s)

Ronald P. Barry

References

Ronald P. Barry, Julie McIntyre. Estimating animal densities and home range in regions with irregular boundaries and holes: A lattice-based alternative to the kernel density estimator. Ecological Modelling 222 (2011) 1666-1672. <doi:10.1016/j.ecolmodel.2011.02.016>

Julie McIntyre, Ronald P. Barry (2018) A Lattice-Based Smoother for Regions with Irregular Boundaries and Holes. Journal of Computational and Graphical Statistics. In Press.

Examples

```
data(nparExample)
attach(nparExample)
plot.new()
#  Simulate a response variable
index1 <- (grid2[,2]<0.8)|(grid2[,1]>0.6)
Z <- rep(NA,length(grid2[,1]))
n1 <- sum(index1)
n2 <- sum(!index1)
Z[index1] <- 3*grid2[index1,1] + 4 + rnorm(n1,0,sd=0.4)
Z[!index1] <- -2*grid2[!index1,1] + 4 + rnorm(n2,0,sd=0.4)
#
plot(polygon2,type="n")

polygon(polygon2)
points(grid2,pch=19,cex=0.5,xlim=c(-0.1,1))
text(grid2,labels=round(Z,1),pos=4,cex=0.5)
#
nodeFillingOutput <- nodeFilling(poly=polygon2, node_spacing=0.025)
plot(nodeFillingOutput)
```

```
formLatticeOutput <- formLattice(nodeFillingOutput)
plot(formLatticeOutput)
hold <- crossvalNparReg(formLatticeOutput,Z,
                       PointPattern=grid2,M=0.5,max_steps = 20)
var_map <- varianceMap(formLatticeOutput,Z,
             PointPattern=grid2,M=0.5,k=10)
plot(var_map)
```

Index